

FIG. 7 is a block diagram of the software environment under which the wireless interface device and the host computer operate to provide remote control of the host computer;

FIG. 8 is a block diagram which shows in further detail the software environment in the host computer, running an application program under a Windows environment;

FIG. 9 is a block diagram which shows in further detail the software environment in the wireless interface device, running in a normal operation state;

FIG. 10 is a block diagram illustrating the method used in the wireless interface device to anticipate a pen/mouse mode decision;

FIGS. 11-30 are schematic diagrams of the wireless interface device in accordance with the present invention;

FIGS. 31-35 are flow charts relating to mouse emulation with a passive pen;

FIG. 36 is a plan view of the wireless interface device illustrating the hot icon area and viewing area of the display;

FIG. 37 illustrates the hot icons in the hot icon area of the display;

FIGS. 38, 39 and 40 are flow charts relating to a system for disabling the screen saver to reduce LAN traffic;

FIG. 40A is a flow chart relating to a host access protection password system;

FIGS. 41-43 are flow charts relating to a system for handling pen-up events;

FIG. 44 is a configuration diagram illustrating the wireless interface device interfacing with a wired LAN system;

FIG. 45 is a diagram of the software structure of a known network system;

FIG. 46 is a diagram of the software structure of network system which enables the wireless interface device to interface with the wired LAN system, illustrated in FIG. 44;

FIGS. 47-52 are flow charts relating to the seamless integration of wired and wireless LANS;

FIGS. 53-57 are illustrations of various set-up dialog boxes available on the wireless interface device;

FIG. is a flow chart relating to the host control mode;

FIG. 59 is a flow chart relating to a system for broadcasting for available hosts;

FIGS. 60 and 61 are flow charts relating to a system for providing remote keyboard macros on the wireless interface device;

FIGS. 62A-62C, 63A and 63B are flow charts relating to a wireless flash memory device programmer;

FIGS. 64A and 64B are flow charts relating to a system for providing automatic reconnection of the host;

FIGS. 65A and 65B are flow charts relating to providing a remote occlusion region on the wireless interface device; and

FIGS. 66A-66D illustrate the various configurations of an on-screen keyboard available on the wireless interface device.

FIG. 67 is a block diagram of the hardware configuration for a system for interfacing multiple wireless interface devices to a single server in accordance with the present invention.

FIG. 68 is a block diagram illustrating the software architecture of the server illustrated in FIG. 67.

FIG. 69 is an overall diagram of the software for wireless enumeration of the server.

FIG. 70 is a view of a dialog box on the wireless interface device in a set-up mode.

FIGs. 71A-71C are flow charts of the software for the wireless interface device for wireless enumeration of servers in accordance with the present invention.

FIG. 72 is a flow chart of the software at the server side for the installation of the server side software for wireless enumeration of the servers available for connection to the wireless interface devices in accordance with the present invention.

FIG. 73 is a flow chart for the software on the server side for providing wireless enumeration in accordance with the present invention.

FIG. 74 is a flow chart for the software on the server side for providing wireless enumeration in accordance with the present invention.

FIG. 75 is an overall flow chart for compressing and decompressing files in accordance with the present invention.

FIGs. 76a-76b are flow charts for compressing .EXE and .COM files in accordance with the present invention.

FIG. 77 is a flow chart for decompressing .EXE and .COM files in accordance with the present invention.

FIG. 78 is a block diagram of an exemplary customized file in accordance with the present invention.

FIG. 79 is a block diagram illustrating an input file and an output file.

FIGs. 80-85 are flow charts for enabling the FLASH memory device on multiple wireless interface devices to be updated wirelessly.

FIGs. 86 and 87 are flow charts for an audio compression system in accordance with the present invention.

FIG. 88 is a graphical representation of an exemplary audio signal.

FIG. 89A is a simplified block diagram of the system illustrated in

FIG. 67, illustrating the speaker and microphone on wireless interface device for running multimedia applications.

FIG. 89B is a block diagram of an audio subsystem in accordance with the present invention.

FIGs. 90-94 are flow charts for a multi-user on-screen keyboard in accordance with the present invention.

FIG. 95 is a simplified diagram illustrating a plurality of overlapping windows and the on-screen keyboard on a display.

FIG. 96 illustrates a container supported by various application programs, such as VISUAL BASIC with an ink field.

FIG. 97 illustrates a data flow diagram for a system for providing ink trails on a wireless interface device in accordance with the present invention.

FIGs. 98-109 represent flow charts for the invention illustrated in FIG. 97.

FIGs. 110-112 represent flow charts for a local handwriting recognition system in accordance with the present invention.

DETAILED DESCRIPTION OF THE INVENTION

1. General

The present invention relates to a system which allows wireless access and control of a remote host computer, which may be either a desktop, tower or portable computer to enable remote access of the various files and programs on the host computer. The system not only allows access to remote host computers that are configured as stand-alone units but also provides access to both wired and wireless local area networks (LAN).

The system includes a wireless interface device which includes a graphical user interface (GUI) which allows various types of input. In particular, input to the wireless interface device is primarily by way of a passive stylus, which can be used in a pen mode or a mouse mode.

As mentioned above, the wireless interface device 100 can also be used with a wireless LAN in a peer-to-peer network or a wired LAN. FIG. 2 illustrates the communication between the wireless interface device 100 and a wired LAN 114, which includes a server 108 in a, for example, Novell Netware or Microsoft LAN Manager environment. In this mode, the transceiver 116 in the wireless interface device 100 communicates with an access point 109 by way of a transceiver (not shown), which interfaces the wireless interface device 100 with a wired LAN 114 which includes a server 108. Alternatively, the wireless interface device 100 can be used in a wireless network in a Windows for Workgroups or Personal Netware environment, for example.

The configuration of the radio communication subsystem between the wireless interface device 100 and the remote host computer 101 or access point 109 conforms to the Open System Interconnection (OSI) reference model for data communications and implements the lower two layers of the seven-layer OSI model. In particular, with reference to FIG. 3, the physical layer 107 (WIRELESS PHY) may be a 2.4 GHz spread spectrum frequency hopping radio which replaces the LAN cable normally connected between workstations. The radio operates within the 2.4000-2.4835 GHz band, the unlicensed Industrial Scientific and Medical (ISM) band, and is divided into eighty-two 1 MHz channels. In a spread-spectrum, frequency-hopping radio, data is broadcast on one particular channel for a predetermined time (i.e. 400 msec); and then the system hops to another channel in a predetermined pattern to avoid interference.

The wireless media access control (WIRELESS MAC) 106 is used to interface to higher level software 105 (i.e. NOS SHELL, NOVELL, MICROSOFT) through network drivers 104 (i.e. LINK LEVEL INTERFACE (ODI, NDIS)). The MAC conforms to the industry standard protocol is in accordance with IEEE 802.11.

Computer 101. Because of local inking, processing by the host computer 101 within the human perceptual response time is rendered unnecessary. Thus, in the host computer 101, the pen events can be queued at pen event buffer 264, to be retrieved one at a time by pen event injector 274. Hence, when pen event buffer 264 is suitably sized, data loss due to overflow by RC buffer 275 is prevented.

Alternatively, the control program 282 can also be implemented to follow a "retractable ball-point pen" paradigm. Under this paradigm, the user controls a local stylus mode of the stylus, such that inking occurs when the stylus is set to be in the local pen mode, and no inking occurs when the stylus is in the local mouse mode. If the local stylus mode conforms with the mode expected by Pen Windows, the image seen on the LCD display of the video display subsystem 113 is the same as described above with respect to state 287 of the control program 282. If the local stylus mode is the mouse mode, and Pen Windows software expects stylus 110 to be in the pen mode, the subsequent video events from host computer 101 would provide the required inking. Finally, the local stylus mode is the pen mode and Pen Windows software expects the stylus to be in the mouse mode, inking would be left on the screen of video display subsystem 113. Under this paradigm, the user would eliminate the erroneous inking by issuing a redraw command to Pen Windows.

5. Detailed Description of the Schematic Diagrams

One embodiment of the invention is illustrated in the schematic drawings, FIGS. 11-30. Referring to FIG. 11 the system may include a CPU 112, such as an AMD. Model No. AM386DXLV microprocessor. The CPU 112 includes a 32-bit data bus D[0..31] as well as a 32-bit address bus A[2..31]. Both the data bus D[0..31] as well as the address bus A[2..31] are connected to the processor bus 150 (FIG. 4), for example, an AT bus. As will be discussed in more detail below, the system controller 129 (FIG. 4) performs

various functions including management of the processor bus 150. In order to conserve power, a 3-volt microprocessor may be used for the CPU 112. As such, a 3-volt supply 3V CPU is applied to the power supply VCC pins on the CPU 112. The 3-volt supply 3V CPU is available from a DC-to-DC converter 300 (FIG. 26) by way of a ferrite bead inductor 302. In particular, the DC-to-DC converter 300 includes a 3-volt output, 3V_CORE. This output, 3V_CORE, is applied to the ferrite bead inductor 302 and, in turn, to the power supply pins VCC of the CPU 112. In order to prevent noise and fluctuations in the power supply voltage from affecting operation of the CPU 112, the power supply voltage 3V CPU is filtered by a plurality of bypass capacitors 304 through 330.

The 3-volt supply 3V_CPU is also used to disable unused inputs as well as to pull various control pins high for proper operation. For example, the 3-volt power supply 3V_CPU is applied to the active low N/A and BS16 pins of the CPU 112 by way of a pull-up resistor 332. In addition, the signals BE[0..3] , W/R, D/C, M/IO and ADS are pulled up by a plurality of pull-up resistors 334 through 348.

The CPU 112 is adapted to operate at 25 megahertz (MHz) at 3.0 volts. A 25 MHz clock signal, identified as CPU CLK, available from a clock generator 398 (FIG. 13), is applied to a clock input CLK2 on the CPU 112 by way of a resistor 349 and a pair of capacitors 351 and 353. The AMD Model No. AMD386DXLV microprocessor supports a static state, which enables the clock to be halted and restarted at any time.

The wireless interface device 100 includes a speaker 355. The speaker 355 is under the control of the system controller 129 (FIG. 12). In particular, a speaker control signal SPKR from the system controller 129 is applied to a source terminal of a field-effect transistor (FET) 357 for direct control of the speaker 355. The drain terminal is connected to the speaker 355 by way of a